UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/870,621	05/31/2001	Scott J. Broussard	AUS920010263US1	1785

35617 7590 01/15/2004

CONLEY ROSE, P.C.
P.O. BOX 684908
AUSTIN, TX 78768

EXAMINER	
BONSHOCK, DENNIS G	
ART UNIT	PAPER NUMBER
2173	3

DATE MAILED: 01/15/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/870,621	**Applicant(s)** BROUSSARD, SCOTT J.	
	Examiner Dennis G Bonshock	**Art Unit** 2173	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *31 May 2001*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-25* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-25* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

13)□ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

a) □ The translation of the foreign language provisional application has been received.

14)□ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) *2* .

4) □ Interview Summary (PTO-413) Paper No(s). _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: .

DETAILED ACTION

Specification

1. The disclosure is objected to because of the following informalities: On page 41, in the paragraph labeled by the number 2, the statement reads "If the background color for the control is not explicitly declared and AWTSwing attempts to get the color from the Swing settings." This statement is in the form of an if-then statement where the then or assumed then is missing.

Appropriate correction is required.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over WinZip computing Inc., WINZIP 8.0., hereinafter WinZip and Java.

4. With regard to claim 1, WinZip teaches, on pages 3 and 4 a system of software components adapted to display text running under an operating system, in which selection of the mask password check box displays the text with one or two software components namely masked ("*********") or unmasked ("password"), the selection of which is made at runtime. WinZip, however, doesn't teach that the appearance of the display is made during runtime. Java teaches a system of masking passwords similar to that of WinZip using the Swing API and namely the JPasswordField and JTextField

(see Java Platform 1.2 Beta 4 API Specification: Class JPasswordField and Class JTextField), but further teaches a system independent display (see IS page 1 paragraphs 1 and 5). It would have been obvious to one of ordinary skill in the art, having the teachings of WinZip and Java before him at the time the invention was made to modify the text display system of WinZip to include the system independence of Java. One would have been motivated to make such a combination because this would allow people on different platforms to use similar and familiar interfaces.

5. With regard to claims 2 and 14, which teach the object is part of a graphical user interface associated with the application program, WinZip teaches, on pages 3 and 4, that the object is part of a GUI associated with the application program WinZip.

6. With regard to claims 3 and 15, which teach the selection of the first or second of the systems of software components being made by the application, WinZip teaches, on pages 3 and 4, that selection of the software used to display the text is made in the application program WinZip.

7. With regard to claim 4 and 16, which teach that an instance of the first or second of the system of software components is created during runtime, when selected by the application program, WinZip teaches, on pages 3-6, that selection of the software used to display the newly typed password is made in the application program WinZip at runtime.

8. With regard to claims 5 and 17, which teach an instance of the first or second of the system of software components previously selected by the application program is destroyed when the instance is no longer selected, WinZip teaches, on pages 3 and 4,

that selection of the software used to display the text is made in the application program WinZip through the use of the checkbox, and if there is existing text in the box when selection is made it will be destroyed and the new selected format will be displayed.

9. With regard to claims 6 and 18, which teach the appearance and behavior of the object differ depending on whether the object is displayed by the first or second of the system of software components, WinZip further teaches, on page 3 and 4, that the appearance is different whether or not the mask is checked. WinZip also teaches, on page 6, that if the password is masked the user will be required to type it twice.

10. With regard to claims 7 and 19, which teach that the application program is written in Java programming language, WinZip teaches the system for displaying the text in two different formats, but doesn't teach the use of Java programming language. Java teaches a system of masking password similar to that of WinZip, but further teaches the use of Java programming language, specifically the Swing API and namely the JPasswordField and JTextField (see Java Platform 1.2 Beta 4 API Specification: Class JPasswordField and Class JTextField). It would have been obvious to one of ordinary skill in the art, having the teachings of WinZip and Java before him at the time the invention was made to modify the text display system of WinZip to use Java. One would have been motivated to make such a combination because this would allow people on different platforms to use similar and familiar interfaces.

11. With regard to claims 8 and 20, which teach the set of software components comprising the Swing application program interface (API), WinZip teaches the system for displaying the text in two different formats, but doesn't teach the use of Swing API.

Java teaches a system of masking passwords similar to that of WinZip but further teaches the use of Java's Swing API and namely the JPasswordField and JTextField (see Java Platform 1.2 Beta 4 API Specification: Class JPasswordField and Class JTextField). It would have been obvious to one of ordinary skill in the art, having the teachings of WinZip and Java before him at the time the invention was made to modify the text display system of WinZip to use Java's Swing API. One would have been motivated to make such a combination because this would allow people on different platforms to use similar and familiar interfaces.

12. With regard to claims 9 and 22, which teach the operating system comprises Windows, Unix or OS/2 computer operating system, WinZip teaches the system for displaying the text in two different formats, but doesn't teach the operating system comprising Windows, Unix or OS/2 computer operating system. Java teaches a system of masking password similar to that of WinZip but teaches the use of Java's Swing API which can be displayed with the same look and feel on Windows, Unix or Apple computers through the use of a components set code-named *Metal (see IS page 1, paragraphs 1- 5).* It would have been obvious to one of ordinary skill in the art, having the teachings of WinZip and Java before him at the time the invention was made to modify the text display system of WinZip to use Java's Swing API for platform independence. One would have been motivated to make such a combination because this would allow people on different platforms to use similar and familiar interfaces.

13. With regard to claims 10 and 21, which teach the set of software components comprising the Swing application program interface (API), WinZip teaches the system

for displaying the text in two different formats, but doesn't teach the use of a TextField and PasswordField. Java teaches a system of masking passwords similar to that of WinZip but teaches the use of Java's Swing API and namely the JPasswordField and JTextField (see Java Platform 1.2 Beta 4 API Specification: Class JPasswordField and Class JTextField). It would have been obvious to one of ordinary skill in the art, having the teachings of WinZip and Java before him at the time the invention was made to modify the text display system of WinZip to use Java's Swing API. One would have been motivated to make such a combination because this would allow people on different platforms to use similar and familiar interfaces.

14. With regard to claims 11 and 23, which teach the selection of either the first or second of the systems software components depends on the status of a software flag associated with the object, WinZip further teaches on page 3 and 4, that the appearance is different whether or not the mask is checked. Java Platform 1.2 Beta 4 API Specification: Class JPasswordField also teaches this in paragraph 1.

15. With regard to claims 12 and 24, which teach that the object is adapted to respond to text entry events and wherein the status of the software flag indicates whether or not a special character is echoed when text is entered, WinZip further teaches on page 3 and 4, that the appearance is different whether or not the mask is checked (characters are displayed if unselected, asterisks are displayed if selected). Java Platform 1.2 Beta 4 API Specification: Class JPasswordField also teaches this in paragraph 1 specifically mentioning an echoChar.

16. With regard to claim 13, WinZip teaches, on pages 3 and 4 a system of software components adapted to display text running under an operating system, in which selection of the mask password check box displays the text with one or two software components namely masked ("********") or unmasked ("password"), the selection of which is made at runtime. WinZip further teaches typing in a character set with the first software component, monitoring the mask flag, and then changing the mode of use of the object upon selection of the flag with an appearance distinct from the first (see pages 3 and 4), and also that if the mask is selected the password must be typed twice (see page 6). WinZip however doesn't teach that the appearance of the display is made during runtime. Java teaches a system of masking passwords similar to that of WinZip using the Swing API and namely the JPasswordField and JTextField (see Java Platform 1.2 Beta 4 API Specification: Class JPasswordField and Class JTextField), but further teaches a system independent display (see IS page 1 paragraphs 1 and 5). It would have been obvious to one of ordinary skill in the art, having the teachings of WinZip and Java before him at the time the invention was made to modify the text display system of WinZip to include the system independence of Java. One would have been motivated to make such a combination because this would allow people on different platforms to use similar and familiar interfaces.

17. With regard to claim 25, WinZip teaches, on pages 3 and 4 a windows based operating system with a system of software components adapted to display text in an application program, in which selection of the mask password check box displays the text with one or two software components namely masked ("********") or unmasked

("password"), the selection of which is made at runtime. WinZip further teaches typing in a character set with the first software component, monitoring the mask flag, and then changing the mode of use of the object upon selection of the flag with an appearance distinct from the first (see pages 3 and 4), and also that if the mask is selected the password must be typed twice (see page 6). WinZip however doesn't teach that the appearance of the display is made during runtime. Java teaches a system of masking passwords similar to that of WinZip using the Swing API and namely the JPasswordField and JTextField (see Java Platform 1.2 Beta 4 API Specification: Class JPasswordField and Class JTextField), but further teaches a system independent display (see IS page 1 paragraphs 1 and 5). It would have been obvious to one of ordinary skill in the art, having the teachings of WinZip and Java before him at the time the invention was made to modify the text display system of WinZip to include the system independence of Java. One would have been motivated to make such a combination because this would allow people on different platforms to use similar and familiar interfaces.

Conclusion

18. The prior art made of record on form PTO-892 and not relied upon is considered pertinent to applicant's disclosure. Applicant is required under 37 C.F.R. § 1.111(c) to consider these references fully when responding to this action. The documents cited therein teach a system of software components in which the components display visually independent objects.

19. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dennis G. Bonshock whose telephone number is (703)305-4668. The examiner can normally be reached on Monday - Friday, 8:30 a.m. - 5:00 p.m.

20. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cabeca can be reached on (703)308-3116. The fax phone number for the organization where this application or proceeding is assigned is (703)746-7239.

21. Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703)305-3900.

dgb

JOHN CABECA
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100